

Do NOT write on this test. Record all answers on the bubble sheet. **Closed book. No notes.** Work strictly from memory. **No calculators. No time limit. Scratch paper okay.**

On the following printf questions you are given a list of inputs. For each problem line determine which printf statement created the accompanying outputs. (   means space.)

Which of these printf statements created the outputs shown for each problem below? (x is int x;)

- (A) printf("\_\_\_\_%-3d",x);    (D) printf("\_\_\_\_%0\_3d\_\_\_\_",x);    (G) printf("\_\_\_\_%-5d",x);  
 (B) printf("\_\_\_\_%\_04d",x);    (E) printf("\_\_\_\_%0+3d\_\_\_\_",x);    (H) printf("\_\_\_\_%\_6d",x);  
 (C) printf("\_\_\_\_%\_d\_\_\_\_",x);    (F) printf("\_\_\_\_%02d\_\_\_\_",x);    (I) printf("\_\_\_\_%-6d",x);

inputs:	<u>  2  </u>	<u>  -9  </u>	<u>1127142423</u>	<u>-2023617651</u>
1/2p.	__2____	__-9____	__1127142423__	__-2023617651__
2/2p.	__02__	__-9__	__1127142423__	__-2023617651__
3/2p.	___2___	___-9___	___1127142423___	___-2023617651___
4/2p.	__+02__	__-09__	__+1127142423__	__-2023617651__

Which of these printf statements created the outputs shown for each problem below? (x is char \* x;)

- (A) printf("\_\_\_\_%2s",x);    (D) printf("\_\_\_\_%1s\_\_\_\_",x);    (G) printf("\_\_\_\_%6s",x);  
 (B) printf("\_\_\_\_%-2s",x);    (E) printf("\_\_\_\_%s\_\_\_\_",x);    (H) printf("\_\_\_\_%-7s",x);  
 (C) printf("\_\_\_\_%\_-4s",x);    (F) printf("\_\_\_\_%\_-5s",x);    (I) printf("\_\_\_\_%7s",x);

inputs:	<u>""</u>	<u>"p"</u>	<u>"bv"</u>	<u>"plgd"</u>	<u>"bhvfd"</u>	<u>"fvgyxkqk"</u>
5/2p.	____	____p____	____bv____	____plgd____	____bhvfd____	____fvgyxkqk____
6/2p.	____	____pp____	____bv____	____plgd____	____bhvfd____	____fvgyxkqk____
7/2p.	____	____p____	____bv____	____plgd____	____bhvfd____	____fvgyxkqk____
8/2p.	____	____pp____	____bv____	____plgd____	____bhvfd____	____fvgyxkqk____

Which of these printf statements created the outputs shown for each problem below? (x is double x;)

- (A) printf("\_\_\_\_%+12.4f",x);    (D) printf("\_\_\_\_%012.2f\_\_\_\_",x);    (G) printf("\_\_\_\_%013.6f\_\_\_\_",x);  
 (B) printf("\_\_\_\_%011.4f\_\_\_\_",x);    (E) printf("\_\_\_\_%\_11.0f\_\_\_\_",x);    (H) printf("\_\_\_\_%014.2f\_\_\_\_",x);  
 (C) printf("\_\_\_\_%+013f",x);    (F) printf("\_\_\_\_%+014f",x);    (I) printf("\_\_\_\_%10.6f\_\_\_\_",x);

inputs:	<u>  3  </u>	<u>-5.74</u>	<u>-4.3809</u>	<u>-393142.722270</u>
9/2p.	000003.000000__	-00005.740000__	-00004.380900__	-393142.722270__
10/2p.	____3____	____-6____	____-4____	____-393143____
11/2p.	__3.000000____	__-5.740000____	__-4.380900____	__-393142.722270____
12/2p.	0000000003.00	-0000000005.74	-0000000004.38	-0000393142.72

Precedence: What is the value of each expression? Mark (I) for error, (J) for none of the above.

13/1p.	$2\%9/4-8/7$	(A) -43	(B) -7	(C) -2	(D) -1	(E) 0	(F) 2	(G) 32	(H) 96
14/1p.	$1-8\%1*4*4$	(A) -50	(B) -39	(C) 0	(D) 1	(E) 4	(F) 9	(G) 16	(H) 40
15/1p.	$2>=4  8+6*6$	(A) -99	(B) 0	(C) 1	(D) 6	(E) 37	(F) 42	(G) 79	(H) 96
16/1p.	$1/2-5\%4*3$	(A) -5	(B) -1	(C) 0	(D) 3	(E) 7	(F) 9	(G) 11	(H) 15
17/1p.	$2-9/1/5*6$	(A) -20	(B) -12	(C) -4	(D) -1	(E) 2	(F) 4	(G) 6	(H) 54
18/1p.	$9\%8\%3/2-4$	(A) -22	(B) -3	(C) -2	(D) -1	(E) 0	(F) 41	(G) 48	(H) 56
19/1p.	$2/1*2+6*3$	(A) -99	(B) 19	(C) 21	(D) 22	(E) 30	(F) 40	(G) 64	(H) 97
20/1p.	$9/8/2+3+8$	(A) 0	(B) 8	(C) 9	(D) 11	(E) 13	(F) 17	(G) 82	(H) 90
21/1p.	$4/4-0*0-4$	(A) -53	(B) -35	(C) -4	(D) -3	(E) 1	(F) 5	(G) 32	(H) 78
22/1p.	$8+1/9+6/4$	(A) -47	(B) 1	(C) 2	(D) 3	(E) 9	(F) 35	(G) 52	(H) 93
23/1p.	$3+5/5/5+1$	(A) -55	(B) 0	(C) 1	(D) 4	(E) 5	(F) 9	(G) 61	(H) 94
24/1p.	$8+4/6\%5+2$	(A) -85	(B) 1	(C) 4	(D) 5	(E) 8	(F) 9	(G) 14	(H) 66
25/1p.	$3*9  0\&\&8-3$	(A) -95	(B) -78	(C) -19	(D) -6	(E) -2	(F) 3	(G) 77	(H) 99
26/1p.	$9-9\%1+4/4$	(A) -54	(B) -9	(C) 1	(D) 3	(E) 8	(F) 9	(G) 10	(H) 52
27/1p.	$4-7-2<=9!=8$	(A) -70	(B) -34	(C) -4	(D) -2	(E) 3	(F) 4	(G) 42	(H) 87
28/1p.	$7\%5>=0<2+4$	(A) 0	(B) 1	(C) 2	(D) 4	(E) 14	(F) 24	(G) 70	(H) 89
29/1p.	$8-7*9-5/8$	(A) -55	(B) -30	(C) -8	(D) -7	(E) 0	(F) 1	(G) 5	(H) 8
30/1p.	$2==2\&\&5\%6-8$	(A) -38	(B) -8	(C) -7	(D) -1	(E) 0	(F) 1	(G) 36	(H) 74
31/1p.	$2+9\&\&4>8-1$	(A) -87	(B) -1	(C) 0	(D) 1	(E) 2	(F) 3	(G) 14	(H) 72

How many times does the body of the loop execute? (Mark 9 if 9 or more.)

32/2p.	<code>int y; for( y=1; y&gt;=2; --y ) body;</code>
33/2p.	<code>int h; for( h=1; h&gt;-3; --h ) body;</code>
34/2p.	<code>int e; for( e=0; e&gt;-4; --e ) body;</code>
35/2p.	<code>int m; for( m=6; m!=9; ++m ) body;</code>
36/2p.	<code>int z; for( z=1; z&lt;4; z++ ) body;</code>
37/2p.	<code>int a=10; while( a++ &lt;= 17 ) body;</code>
38/2p.	<code>int e=8; do body; while( --e != 6 );</code>
39/2p.	<code>int x=3; while( x-- != 3 ) body;</code>

40/3p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.

Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^2\sqrt{n}$  (C)  $n^2$  (E)  $n\sqrt{n}$  (G)  $n$  (I)  $\sqrt{n}$   
 (B)  $n^2\lg n$  (D)  $n\sqrt{n}\lg n$  (F)  $n\lg n$  (H)  $\sqrt{n}\lg n$  (J)  $\lg n$

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        i = 1; while ( i * i < n ) {
            simpleStatement;
            i += 1; }
    } else {
        simpleStatement;
    }
    return 0; }
```

41/3p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.

Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^2\sqrt{n}$  (C)  $n^2$  (E)  $n\lg n$  (G)  $\sqrt{n}\lg n$  (I)  $\lg n$   
 (B)  $n^2\lg n$  (D)  $n\sqrt{n}$  (F)  $n$  (H)  $\lg^2 n$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        d = 1; while ( d * d < n ) {
            for ( i = n ; i > 1 ; i /= 2 ) {
                simpleStatement;
            }
            d++; }
    } else {
        simpleStatement;
    }
    return 0; }
```

42/3p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.

Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^2\sqrt{n}$  (C)  $n^2$  (E)  $n$  (G)  $\sqrt{n}$  (I)  $\lg n$   
 (B)  $n^2\lg n$  (D)  $n\sqrt{n}$  (F)  $\sqrt{n}\lg n$  (H)  $\lg^2 n$  (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    if ( simpleCompare ) {
        i = 1; while ( i * i < n ) {
            j = 1; while ( j * j < n ) {
                simpleStatement;
                j++; }
            i += 2; }
    } else {
        simpleStatement;
    }
    return 0; }
```

- 43/5p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.  
 Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.
- (A)  $n^2\sqrt{n}$  (C)  $n\sqrt{n}\lg n$  (E)  $n\lg n$  (G)  $\sqrt{n}\lg n$  (I)  $\lg^2 n$   
 (B)  $n^2$  (D)  $n\sqrt{n}$  (F)  $n$  (H)  $\sqrt{n}$  (J) 1

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  if ( simpleCompare ) {
    if ( simpleCompare ) {
      if ( simpleCompare ) {
        h = 1; do {
          simpleStatement;
          h *= 2; } while ( h < n );
        }
      } else {
        simpleStatement;
      }
    } else {
      i = 1; while ( i * i < n ) {
        simpleStatement;
        i++; }
    }
  }
  return 0; }
```

- 44/5p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.  
 Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.
- (A)  $n^4$  (C)  $n^2\sqrt{n}$  (E)  $n^2$  (G)  $n\sqrt{n}\lg n$  (I)  $\sqrt{n}\lg^3 n$   
 (B)  $n^3\lg n$  (D)  $n^2\lg n$  (F)  $n\sqrt{n}\lg^2 n$  (H)  $n\lg^3 n$  (J)  $\lg n$

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  if ( simpleCompare ) {
    j = 1; while ( j < n ) {
      f = n; while ( f > 1 ) {
        e = 1; do {
          for ( g = 1 ; g * g < n ; g++ ) {
            simpleStatement;
          }
          e *= 2; } while ( e < n );
        f -= 2; }
      j *= 2; }
    } else {
      k = 1; while ( k * k < n ) {
        if ( simpleCompare ) {
          simpleStatement;
        }
        k += 3; }
    }
  }
  return 0; }
```

45/5p. Give a tight big-oh  $\Theta()$  bound on the running time  $T(n)$  of this program.

Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in  $\Theta(1)$  time.

- (A)  $n^4\sqrt{n}$  (C)  $n^3\lg^2 n$  (E)  $n\sqrt{n}\lg^3 n$  (G)  $n\lg^4 n$  (I)  $\lg^2 n$   
 (B)  $n^4$  (D)  $n^2\sqrt{n}$  (F)  $n\sqrt{n}$  (H)  $n\lg^3 n$  (J) 1

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  b = n; while ( b > 1 ) {
    f = 1; while ( f * f < n ) {
      if ( simpleCompare ) {
        if ( simpleCompare ) {
          for ( g = 1 ; g < n ; g *= 5 ) {
            if ( simpleCompare ) {
              if ( simpleCompare ) {
                simpleStatement;
              } else {
                simpleStatement;
              }
            } else {
              simpleStatement;
            }
          }
        } else {
          if ( simpleCompare ) {
            d = 1; do {
              simpleStatement;
            } while ( d * d < n );
          } else {
            simpleStatement;
          }
        }
      } else {
        if ( simpleCompare ) {
          a = n; while ( a > 1 ) {
            for ( j = n ; j > 1 ; j /= 2 ) {
              i = n; do {
                simpleStatement;
              } while ( i > 1 );
            }
            a /= 5; }
          } else {
            if ( simpleCompare ) {
              simpleStatement;
            } else {
              simpleStatement;
            }
          }
        }
      }
    }
  }
  f++; }
  b /= 2; }
  return 0; }
```

Matching: Which Perl regular expression commands have what meaning? (If no match mark J.)

(A) , (B) \B (C) \S (D) \b (E) \n (F) \r (G) \s (H) {n,m} (I) {n..m}

- 46/1p. at least n times, at most m
- 47/1p. or (alternation)
- 48/1p. carriage return
- 49/1p. whitespace
- 50/1p. not a word boundary

True or False: does the string match the regular expression?

- 51/1p. Does the string "rqq" match the regular expression "rq\*|zd"?
- 52/1p. Does the string "uzuzru" match the regular expression "uz?ru"?
- 53/1p. Does the string "ahah" match the regular expression "(ah)+(ks)?"?
- 54/1p. Does the string "uctw" match the regular expression "uc|tw"?
- 55/1p. Does the string "ausxsxx" match the regular expression "(au\*(sx)+)?"?
- 56/1p. Does the string "yw" match the regular expression "ap\*ay+|yw"?
- 57/1p. Does the string "kkkkn" match the regular expression "(k\*|wr)?n"?
- 58/1p. Does the string "nbbycggg" match the regular expression "(nb)?yc|(g+)?"?
- 59/1p. Does the string "xkxkf" match the regular expression "(xk)+dn\*(f+)+"?
- 60/1p. Does the string "qhrxxt" match the regular expression "n|a|q|hr+|xt"?
- 61/1p. Does the string "bt" match the regular expression "tz|(gu)\*|(bt)\*(zg)\*"?
- 62/1p. Does the string "hghq" match the regular expression "fw|c\*|x|(hq)?"?

---

Total points 100.

**Answer Key** (points per line)

1 (2).	G	32 (2).	0
2 (2).	F	33 (2).	4
3 (2).	A	34 (2).	4
4 (2).	E	35 (2).	3
5 (2).	B	36 (2).	3
6 (2).	G	37 (2).	8
7 (2).	H	38 (2).	2
8 (2).	I	39 (2).	0
9 (2).	G	40 (3).	$I(\sqrt{n})$
10 (2).	E	41 (3).	$G(\sqrt{n} \lg n)$
11 (2).	I	42 (3).	$E(n)$
12 (2).	H	43 (5).	$H(\sqrt{n})$
13 (1).	D (-1)	44 (5).	$F(n\sqrt{n} \lg^2 n)$
14 (1).	D (1)	45 (5).	$E(n\sqrt{n} \lg^3 n)$
15 (1).	C (1)	46 (1).	H
16 (1).	J (-3)	47 (1).	J
17 (1).	C (-4)	48 (1).	F
18 (1).	J (-4)	49 (1).	G
19 (1).	D (22)	50 (1).	B
20 (1).	D (11)	51 (1).	true
21 (1).	D (-3)	52 (1).	false
22 (1).	E (9)	53 (1).	true
23 (1).	D (4)	54 (1).	false
24 (1).	J (10)	55 (1).	false
25 (1).	J (1)	56 (1).	true
26 (1).	G (10)	57 (1).	true
27 (1).	J (1)	58 (1).	false
28 (1).	B (1)	59 (1).	false
29 (1).	A (-55)	60 (1).	false
30 (1).	F (1)	61 (1).	true
31 (1).	C (0)	62 (1).	false

Total points 100.